ISSN 0956-9979

OCTOBER 2001

VIRUS BULLETIN

THE INTERNATIONAL PUBLICATION ON COMPUTER VIRUS PREVENTION, RECOGNITION AND REMOVAL

IN THIS ISSUE:

- **Sing a Rainbow:** We've had Code Red, Code Green and now Adrian Marinescu brings us an analysis of the latest addition to the rainbow of worm colours, BlueCode. See p.6.

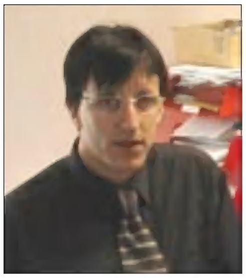
- **Arabian Nights:** Eddy Willems was disconcerted to find his possessions being rifled through by Saudi customs officials, but not as horrified as when he witnessed them inserting virus-infected disks into their unprotected systems. Read the full horror story starting on p.10.

- **Passing the Blame:** Who should be held responsible for the damages caused by malware outbreaks and can we sue any of them? Ray Glath ponders the legalities and shares his opinions on p.16.

CONTENTS

VIRUS BULLETIN ©2001 Virus Bulletin Ltd, The Pentagon, Abingdon, Oxfordshire, OX14 3YP, England.
www.virusbtn.com /2001/$0.00+2.50

COMMENT



“ It is indicative of a general malaise in the IT world, that Code Red has become an AV issue ”

A Worm by Any Other Name …

A worrying trend in the anti-virus world is a sudden upturn in both interest and execution of so-called ‘benevolent’ worms – that is, worms which try to patch the vulnerabilities that other worms exploit.

Code Red has been the most recent, and highest profile victim of this sort of foolishness.

In fact, the Code Red issue is not, or should not be, an anti-virus issue at all.

It is, in this case, not a virus that has caused the problem (although it has certainly highlighted it), nor is it a problem that can be adequately addressed by AV.

It is indicative of a general malaise in the IT world, that Code Red has become an AV issue. People hear the word ‘virus’, and immediately reach for AV without thinking, but why did they have to wait for a worm to discover that they had vulnerable systems? The customers who screamed and begged at their vendor to provide a patch for Code Red, are the same customers who simply ignored the fact that an update that would entirely prevent infection by Code Red on their systems had been available for some weeks before Code Red was even released.

Various forums have been littered with complaints that person X had ‘removed’ Code Red from their system with a product supplied by their AV vendor, only to find that they are now reinfected. Disinfecting Code Red using AV is like treating emphysema with cough syrup – a temporary alleviation of the symptom with nil effect on the underlying disease.

Now we have a few more problems added into the mix. After much discussion in forums such as alt.comp.virus, and focus-virus (Security Focus), it seems that a number of people still thought it was a good idea to treat the Code Red problem by creating further worms that would attempt to disable the initial one.

First came Code Green – posted to Security Focus, an ill-conceived piece of junk that attempts to patch the servers it infects. As the author admits, there’s no guarantee the code even works; and for reasons quite unrelated to its intended functioning, I have to say he’s right. It simply doesn’t work; there is no way that two worms are better than one.

Perhaps more worrying is BlueCode (*for full details see Adrian Marinescu’s analysis, p.6 - Ed.*), which rather than even attempting to patch the vulnerability simply breaks the installation of IIS on the servers it infects. Thus this rather distasteful piece of malware cannot even claim benevolence.

Sadly, this problem isn’t going to go away until the last installation of IIS is patched. How many more colours will it take to convince people of what has always been a black and white issue? Worms that attempt to fix problems created or exploited by other worms are simply an extension of the same problem.

More than two months after the discovery of Code Red, it appears too many people still haven’t got the message. I guess the old saying ‘Bad news is swallow winged, what’s good comes on crutches’, is never more true than when it comes to Internet communication.

A cumulative patch that will fix the vulnerability exploited by Code Red and other exploits, can be found at: http://www.microsoft.com/technet/security/bulletin/MS01-044.asp.

Andrew J Lee
Dorset County Council, UK

NEWS

US Disaster – the Aftermath

VB was shocked and deeply saddened by the recent terrorist attacks in the USA. As might have been predicted, it was not long before the knock-on effects of the atrocities were felt in the information security world, with hoaxers, hackers and virus writers alike taking advantage of the high emotions aroused by the events.

We have been warned of email hoaxes inviting us to give donations to bogus charities in the guise of donating to victims of the attacks, and a number of chain letters have been doing the rounds, ranging from a petition of sympathy to the USA to claims that Nostradamus predicted the terrorist activity in the US. In addition, the US National Infrastructure Protections Centre (NIPC) has alerted companies to be on guard against viruses disguised as files containing information about the attacks. NIPC has reported that a new Life Stages variant dubbed 'WTC' has been spreading, arriving as an attachment to an email purporting to be about the World Trade Center.

The hacking community too has seen a flurry of activity, with 'hacking vigilantes' laying siege to official Taleban Web sites – it appears also that one consortium of hacking groups may have erroneously defaced a Web site operated by one of the companies whose offices were based in the World Trade Center.

On a more constructive note, more than 1000 offers of assistance and equipment were received from the IT community in just 24 hours following an appeal for technical assistance by the New York and Washington DC Red Cross offices. The general message for those wishing to do something positive about the events in the USA is to go through official channels such as the Red Cross Web site at http://www.redcross.org/ ▮

Every Trick in the Book?

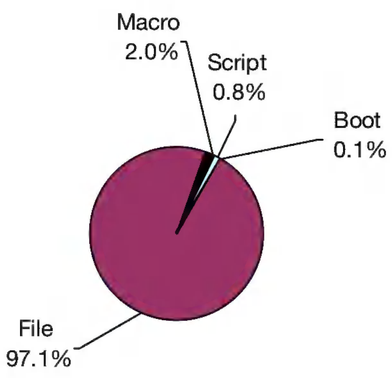
On the tips of everyone's tongues as this issue goes to print is a new mass-mailer which has begun spreading with speed and vigour and is keeping the phone lines hot in AV support departments. Mass-mailer W32/Nimda.A@mm also spreads via network shares, the *Microsoft* Web Folder Transversal vulnerability (which was also used by W32/CodeBlue – see p.6), and a *Microsoft* MIME Header vulnerability in *MS Outlook*, *Outlook Express* and *Internet Explorer*. In addition, the worm attempts to create network shares, and make use of the backdoor created by the W32/CodeRed.c worm as well as opening additional security holes. The worm has been described as a combining the mechanisms of Code Red-II, Kakworm and SirCam – and judging by their respective places in the prevalence table, this will be a powerful combination indeed ▮

Prevalence Table – August 2001

Virus	Type	Incidents	Reports
Win32/SirCam	File	14740	83.5%
Win32/Magistr	File	976	5.5%
Win32/Hybris	File	661	3.7%
Win32/MTX	File	240	1.4%
Win32/Funlove	File	138	0.8%
Win32/CodeRed-II	File	137	0.8%
Laroux	Macro	131	0.7%
Win32/BadTrans	File	83	0.5%
Divi	Macro	54	0.3%
Kak	Script	43	0.2%
Solaris/Sadmind	File	42	0.2%
Haptime	Script	37	0.2%
Marker	Macro	37	0.2%
Win32/QAZ	File	36	0.2%
LoveLetter	Script	35	0.2%
VCX	Macro	27	0.2%
Ethan	Macro	17	0.1%
VBSWG	Script	17	0.1%
Win32/Navidad	File	13	0.1%
Win32/Pretty	File	13	0.1%
Tristate	Macro	12	0.1%
Win32/Kriz	File	12	0.1%
Win32/Ska	File	12	0.1%
Thus	Macro	11	0.1%
Others [1]		136	0.8%
Total		**17660**	**100%**

[1] The Prevalence Table includes a total of 136 reports across 60 further viruses. Readers are reminded that a complete listing is posted at http://www.virusbtn.com/Prevalence/.

Distribution of virus types in reports

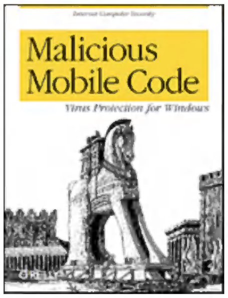


BOOK REVIEW

Reading Between the Worms

Paul Baccas
Sophos Anti-Virus, UK

Malicious Mobile Code: Virus Protection for Windows
Author: Roger A. Grimes
ISBN 1-56592-682-x, Publisher: O'Reilly, Price: $39.95



It seems that most computer books are about 40 mm thick, with approximately one third of that representing 'useful' information. O'Reilly have a reputation, as publishers, for distilling their texts so that only the 'useful' third is published. *Malicious Mobile Code* (*MMC*) is 25mm thick – I would hate to see the tome another publisher would have produced!

The field of computer viruses is not one that has received the attention of serious authors in recent years. In fact, the day before I received my copy of *MMC* I was asked to recommend a book on viruses for a new employee; I was reduced to recommending a book published in 1994. Considering how much has happened since 1994, in terms of technological advances of both viruses and AV software, it is a sad reflection on the number of books available about this subject.

At the time of writing this review, the online errata for the book was empty. This fact surprised me, because the book's editing is not of the standard I have come to expect. Inaccuracies range from imaginary products to non-existent sections of the book and references within 'Chapter X' claiming 'this will be covered in more detail in Chapter X'. By far the most common mistakes are simple grammatical ones. However, none of these faults are disastrous, and a native English speaker will interpret them without problem – the errors simply jar the reader's senses and mar the overall experience of what, in my opinion, is the best book (on viruses in particular and computer security in general) I have seen in a while.

The book is split into two sections (though not by the author). The first section concerns traditional malware: DOS Viruses, Viruses in *Windows*, Macro Viruses, and Trojans and Worms. The second half of the book considers newer and more esoteric subjects: Instant Messaging, Browser Attacks, Malicious Java and ActiveX and Email-Aware Code. Elsewhere, coverage of the subject of the first part of the book can be found in various disparate places, and that of the second part can normally be found in literature relating to security or security exploits. Yet, here the two are presented together with clarity, accuracy (in the technical respect) and, above all, without the accompanying flotsam one would normally associate with a book of this nature.

Each of the chapters has a similar feel. Each begins with a discussion of the technologies affected by the malicious mobile code. Next, the sub-types of the particular malicious code are discussed, and examples are given. For each facet, the most useful part for most people will be the discussions on detection of, removal of and protection from the malicious code. The final part of each chapter is concerned with a current and future risk assessments.

Roger A. Grimes obviously has a deep understanding of his subject and the book is pitched at a level to educate the advanced home user and IT administrator, to provide pointers to security specialists and to reinforce knowledge that analysts and experts alike have acquired. The chapter on DOS viruses is covered quickly, in 36 pages, but the coverage is thorough and is all that you really need. In fact, I am sure that bits could have been cut from that section were it not for the fact that fundamental concepts of viruses, Trojans and worms are introduced in it.

However, the following two chapters alone make this book worth its purchase price. They are entitled *Windows Technologies* and *Viruses in a Windows World*. What they cover should be obvious, but it is *Windows* that is the current battleground. Over three quarters of viruses in the July 2001 prevalence tables (see *VB* September 2001 p.3) have the 'Win32/' prefix. The part of the book that I suspect will be most widely used is that describing the 'executable path'. When I started in the AV industry, one of the things I remember having to learn about was the 'executable path', a virus being a piece of self-replicating code that placed itself in the 'executable path'. In 1997, the virus writers had not yet discovered the intricacies of the *Windows* 'executable path' (the DOS one being relatively mundane) and now the list is so long that I have to think about what they all are. Or rather I *had* to think what they all are because they are now at my fingertips, or at least in my library.

There is nothing particularly striking or revolutionary in *Malicious Mobile Code*, except for the fact that all this information is combined in one easy-to-access book. Most of what is covered is common sense and experience, however in my estimation those qualities are not often combined. Grimes groks his subject and it looks as if he could have filled another book. So, before I tell you to go out and buy this book, I will have one final gripe: it appeared, to me, that Grimes has a favourite AV product and I found that galling.

Despite some misgivings, the technical content of this book is very good, and I believe that every IT department should consider purchasing a copy.

CONFERENCE PREVIEW

AVAR 2001

Allan Dyer
AVAR 2001 Conference Chairman

Three years ago, a small group of AV researchers met in Hong Kong and linked hands for a photograph. This was the inaugural event of the Association of anti-Virus Asia Researchers, an independent and not-for-profit organization based in the Asia Pacific region.

AVAR – the Concept

AVAR was the brainchild of Seiji Murakami, a leader in Japanese Anti-Virus who developed the first local anti-virus product in 1990 and who founded both the Japan Computer Security Research center (JCSR) and the Japan Computer Security Association (JCSA). Murakami realized that there was a need for an independent, non-profit-making anti-virus organization in Asia. He contacted other researchers around the region and, in June 1998, formed AVAR, whose mission is to prevent the spread of and damage caused by computer viruses, and to develop a cooperative relationship among anti-virus researchers in Asia. Although Asia is the focus of interest and activities, there is no requirement for members or subscribers to be Asian, nor even located in Asia.

There are three levels of membership within AVAR: individual, corporate and subscriber. Individual and corporate applicants must be proposed for membership by a current member, and approved by the Board of Directors. Both members and subscribers receive AVAR mailing lists, which keep everyone in contact – we can be discussing a 'hot' topic before those in more tardy time zones are awake. We have also seen longer-term geographical differences through virus incident reports: although the numbers are now much lower, the reports exchanged show that CIH is still more prevalent in Asia than elsewhere in the world.

AVAR Conference

AVAR's main activity, though, is its annual conference which has seen a steady increase in size since the inaugural event – the second conference, in Korea, attracted 50 participants, while last year's conference, in Tokyo (see *VB* January 2001, p.10), saw this figure rise to 180. This year the conference returns to Hong Kong, and will be held at the New Renaissance Hotel, on 4 and 5 December.

This year's conference is co-organised by the Information Security Special Interest Group (IS-SIG) of the Hong Kong Computer Society (HKCS). The Hong Kong Computer Society was founded in 1970 as a non-profit-making professional body whose primary objective is to promote the use of IT in Hong Kong. The IS-SIG was established in June 2000 and focuses on research and discussion of security-related subjects.

Sponsors of the conference include *Network Associates*, the Information Technology Services Department (ITSD) of the Hong Kong Government, *Symantec*, *Ahnlab*, *VirusBuster* (Hungary), *HAURI* and *Microsoft*. Other supporting organizations include the Hong Kong Information Technology Federation, the Computing Services Centre of City University of Hong Kong, the Singapore Computer Emergency Response Team (SingCERT), Infocomm Development Authority of Singapore (IDA), China's National Computer Virus Emergency Response Centre, Anti-Virus Products Testing and Certification Centre and Taiwan's Chinese Cryptology and Information Security Association.

One unique feature of each of the AVAR conferences has been government involvement, with speakers in previous years including the Korean Information Security Agency (KISA), the Japanese Ministry of International Trade and Industry, the Infocomm Development Authority of Singapore and the Chinese Tianjin Quality Testing and Inspection Service. This year government topics will include Information Security Policy in Japan and the introduction of a National Computer Virus Emergency Response Center in China.

The techies should not feel left out either – two papers look at the future of virus detection in new *Office* versions, while other papers consider the use of Intrusion Detection Systems for catching viruses and the security of Java mobile phones. Sun Tze advised in his lessons on the art of war that one should know the enemy and know oneself, so the papers on how worms can be successful and how best to compare AV software are entirely appropriate.

For the Corporate Security Manager, the presentations on a major corporation's virus checking service, and on grassroots exchange of anti-virus information will be of special interest.

As the Conference Chairman, I would not like to suggest that this short list of topics will be conference highlights. Full programme information and the participation details will be available on the AVAR Web site.

Conference:	AVAR 2001
Dates:	4–5 December 2001
Venue:	New World Renaissance Hotel, Hong Kong.
Web:	http://www.aavar.org/.

VIRUS ANALYSIS

Red Turning Blue

Adrian Marinescu
GeCAD Software, Romania

Shortly after Code Red made its first appearance, it was pretty clear that more worms like this, using exploits in various software components, would be seen in the near future. Sure enough, CodeRed.c, which uses the same exploit as the original release, was designed to affect an even greater number of servers than the original version.

Although not as widespread as Code Red, BlueCode is an interesting piece of malware, and understanding its makeup might give us some clues as to what action needs to be taken against similar worms, that seem likely, if not certain, to be developed during the next few years.

Background

Unicode allows multiple encodings for each character. '/', for instance, can be represented in many ways: 0x2F, 0xC0 0xAF, 0xE0 0x80 0xAF. Normally, the IIS server checks whether the requested object is inside the Web root. Requests such as '../' or '..\' should be denied automatically – gaining access to such objects would give you more than the level of access set by the administrator to the Web server. However, a bug in the validation routine that should reject such URLs causes IIS to use the requested objects, even though they are not public. Using this trick, remote commands can be executed on the affected Web server – one could easily run cmd.exe and do a lot of harm on that server. It wouldn't take much for an evil mind to realize that this is all you need to take over the system.

Fortunately things are not so easy for an inexperienced hacker. Using pipes to redirect input/output for various programs like telnet and ftp causes a server error. Of course, this did not prevent the exploit from being used – programs like tftp (Trivial File Transfer Protocol), which come in the standard *Windows NT*/*Windows 2000* installation, do not require input from the users; the only problem was finding a tftp server from which to download files.

Interestingly, this whole mechanism was described on several security lists last year – it took a year for someone to realize that a worm could be created based on this mechanism.

The Worm

BlueCode is an IIS worm that exploits the 'Web Directory Traversal' vulnerability – a rather old vulnerability which was fixed by *Microsoft* in August, 2000. Unlike its predecessor Code Red, which was a so-called 'fileless' worm, BlueCode replicates using files.

The main worm component, called 'httpext.dll', is an IIS extension, and is a DLL file of about 44Kb, developed using *Microsoft Visual C++*. When invoked, the IIS extension will drop on the disk a file called 'svchost.exe' – an executable file of about 14Kb, compressed with the well-known packing utility UPX 1.20. The third file component of the worm is a dropped VBScript file that is used to make working with several IIS services easier. The filenames were cleverly selected to make the worm's presence less conspicuous in the system: both 'svchost.exe' and 'httpext.dll' are names of genuine files present in the default *Windows*/IIS installation.

The worm's first contact with a system to be infected is a malformed HTTP GET request. Due to the bug in the handling routine of UNICODE strings described earlier, BlueCode tries to download itself via the tftp protocol to the affected machine as an IIS extension named 'httpext.dl''. All that is needed to activate the worm is to open the associated URL for it. When the extension is loaded, it will create a global atom named 'CodeBlue' to make sure only one copy of the worm is active. Then, it will drop a file named 'C:\svchost.exe' and execute it.

Replication

When executed, 'svchost.exe' will first make itself a tftp server. This is accomplished by listening on the local port 69 for incoming connections. BlueCode creates a pool of 100 listeners – which is considered to be sufficient for simultaneous connections. Next, it will attempt to record itself in the registry so that it will be started each time the computer boots up. At this point it will also try to set the 'hidden' and 'system' attributes of its host file.

At this stage, BlueCode will attempt to stop all the problematic IIS extensions known to have exploits (Code Red being one of them). This is why BlueCode has been described by some as a 'good' worm, that prevents others from entering the same server. However, this is pure idiocy, and it should be clear to everybody that this is *not* the case.

BlueCode attempts to drop a file named 'C:\d.vbs'. When executed, this file will attempt to stop the '.ida', '.idq' and '.printer' IIS services (all are known to have several exploits). After that, BlueCode attempts to enumerate the processes in the system and terminate all processes named 'inetinfo.exe' (the IIS standard service filename). This part depends on the operating system: since it uses process-specific APIs implemented only in *Windows 95/98/ME* and *Windows 2000*, it will not work on *WindowsNT* systems.

At this point, a routine that resolves the current host name and its IP is called. Next, BlueCode will attempt to create 100 threads, with a delay of 137 milliseconds between the creation of each. After creating them all, it will pause for five seconds and then delete the temporary file named 'C:\d.vbs'.

Each thread will check the current system time. If the time is between 10am and 11am it will try to make a DoS attack on the host 211.99.196.135 (which formerly pointed to http://www.nsfocus.com/), sending GET requests continuously and recreating the threads.

If the time is not between 10am and 11am, BlueCode will attempt to spread itself. Based on the current time, a random number is generated. Depending on a randomly generated value, in half of the cases, the worm will use the current B class and select a random host from that class. In the rest of the cases, a fully random IP address is selected. BlueCode will connect to that IP address and try to determine whether the remote computer is running IIS (this is done by searching for the string IIS in the response from the server).

Spreading

The spreading mechanism is simple: first, it sends a malformed GET command to the IIS server, which will download an IIS extension, named httpext.dll. Next, a GET command that runs the extension is issued to the server. This way, the IIS extension gains control. That extension will drop the file named 'SVCHOST.EXE' in the root of the C:\ drive and execute it.

In addition, BlueCode acts like a tftp server – which makes uploading the worm to the remote machine very simple: just send a tftp command that will get the IIS extension from the local machine. The tftp command is included by default only in *WindowsNT* and *Windows 2000*.

After uploading itself and invoking the IIS extension, the worm uses the IIS vulnerability for the third time; this time to copy 'httpex.dll' in the root of the 'C:\' drive – that is the path used by the tftp server when serving clients. Because of the type of the vulnerability, only servers with the 'wwwroot' directory on the same partition as the WINDOWS directory are vulnerable.

Fixes

This vulnerability was described in *Microsoft Security Bulletin MS00-078*, which was released in October 2000. The *Microsoft* patch, which is even older (released in August 2000), is available from the following locations: http://www.microsoft.com/ntserver/nts/downloads/critical/q269862/default.asp and http://www.microsoft.com/windows2000/downloads/critical/q269862/default.asp (for both *NT* and *2000*).

On 30 November 2000, *Microsoft* announced a newly-discovered regression error which affected IIS 5.0 systems (*Windows 2000*). The bug made servers vulnerable to the 'Web Server Directory Traversal' exploit even when the recommended patch had been installed. All IIS 5.0 users were encouraged to install a newer patch, available at: http://www.microsoft.com/Downloads/Release.asp?ReleaseID=25547.

Containing the Spread

Many factors limit the spread of BlueCode. First, the mechanism is based on a rather old vulnerability. Second, the method of file transfer is a rather uncommon one, and firewall software should not permit tftp connections unless necessary. My true hope, however, is that system administrators have learned the Code Red lesson and have updated their problematic software components.

Conclusion

BlueCode's replication mechanism is more similar to that of the recent *Linux* worms than that of the Code Red IIS worm. Indeed the concept of the 'patching-worm' was seen in the Linux/Cheese worm, which attempted to patch those security holes used by one of the Linux/Lion worms to replicate.

The idea of 'good viruses' is not a new one – several virus creators have written viruses that have been able to detect and deactivate several other viruses. However, anyone who thinks this is the solution to the malware problem is very wrong – such 'good' malware is no less dangerous than the targeted ones.

Firewall plugins should be developed to prevent worms like this from spreading at such a high rate as Code Red. Moreover, firewall functions should be added into all standard AV protection until reliable heuristic detection methods have been developed for this type of malware.

Finally, system administrators should be more vigilant when it comes to security issues. Both Code Red and BlueCode have made use of known exploits, fixes for which had been available for download from *Microsoft* for a long time.

W32/BlueCode.worm

Aliases:	IIS-Worm.BlueCode, CodeBlue.
Type:	Network-propagated worm.
Infects:	*WindowsNT/2000* machines running unpatched IIS4/5.
Payload:	Between 10am and 11am tries to DoS a security-related site.
Removal:	Stop the IIS service and install the recommended patch from *Microsoft*. Remove C:\SVCHOST.EXE and C:\HTTPEXT.DLL, remove the HKLM\Software\Microsoft\Windows\CurrentVersion\Run\Domain Manager registry key, then start the IIS service.

TECHNICAL FEATURE

Red Number Day

Dmitry Gryaznov
Network Associates Inc., USA

For me, the CodeRed.c (aka Code Red II) story began on Saturday, 4 August 2001. Several days previously, in order to track the 'resurrected' CodeRed.b (aka Code Red v2), I had written and launched a simple program I named 'FakeHTTP' – a fake HTTP server, which listens for incoming connections on port 80, accepting and logging everything it receives and responding to all requests with HTTP error '404 Not found'.

My home computer is connected to the Internet by a cable modem and I had been running my FakeHTTP program from home for a couple of days. It was registering mostly CodeRed.b attempts on my computer at an average rate of approximately one per hour, most of them coming from Korea and China (both PRC and ROC).

Unexplained Activity

When I checked my computer early in the afternoon of 4 August, what caught my attention immediately was the unusually high network activity, as indicated by constantly blinking LEDs on my cable modem. I consulted the FakeHTTP log file and saw numerous entries of something new, which was coming in on average every five to six minutes. It was similar to CodeRed.b, yet instead of the then very familiar GET /default.ida?NNNNNNNN… this one had GET /default.ida?XXXXXX…

Despite it being the weekend, I contacted my fellow virus researchers and we began analysing. Of course, it was CodeRed.c.

First Sightings

According to my FakeHTTP log, the first instance of CodeRed.c arrived at my computer at 6.42am PDT (1.42pm GMT) 4 August from another cable provider network on the US East Coast. I would have thought that it must have been rather widespread by that time, but I am aware of only one earlier sighting of CodeRed.c – also on the West Coast and also coming from the East Coast. It looks as if the virus was 'injected' into the Internet through a network on the East Coast an hour or two before it arrived at my computer.

After a short while, still observing high network activity due to the virus and frequent arrivals of it registered by FakeHTTP, I was curious as to how widespread the virus had become. I did not have access to any large network's logs – all I had was the log from my own single computer. However, due to the way in which the virus spreads, that appeared to be enough.

Virus Algorithm

There have been numerous detailed technical descriptions of CodeRed.c, so I shall not bore you with another. For the purpose of this article only a few details of the virus algorithm are important:

- Like variants .a and .b, CodeRed.c is 'language-aware': the virus checks the system's default language and the virus behaviour varies depending on whether the default language is Chinese (either Traditional or Simplified) or another language. I shall refer to such systems as 'Chinese' and 'non-Chinese', respectively.
- CodeRed.c generates 300 'spreading' threads on a 'non-Chinese' system and 600 such threads on a 'Chinese' one.
- Each 'spreading' thread loops, generating a random IP address and attempting to send the virus to that address. Unlike CodeRed.b, the IP addresses generated by CodeRed.c are not completely random.

The virus 'skews' the probabilities to favour IP addresses of computers 'close' to the attacking one: with a probability of 1/2 the 'random' IP address will have the same upper octet as the attacking computer's IP address. That is, if the infected computer's IP address is 192.168.130.4, in 50 percent of attempts it will probe computers with IP addresses of the form 192.*.*.* (where '*' represents any octet 1 to 254). Such an IP range is often referred to as 'class A network' – while the term is not quite correct in this case, I shall use it for lack of a better one.

With a probability of 3/8, the randomly-generated IP address will have the same two upper octets as the attacking computer's IP address. Continuing with the above example, this would be 192.168.*.*. Such an IP range is often referred to as 'class B network' (again, while I appreciate that this is not strictly the correct use of the term, I shall use it nevertheless).

Finally, with a probability of 1/8, the generated IP address is random.

Estimating Numbers

I made my first estimate of CodeRed.c numbers worldwide on 5 August, about 33 hours after the first sighting of the virus by my FakeHTTP program. To make an estimate such as this, we need first of all to establish how quickly an infected computer probes different IP addresses. In other words, how many IP addresses it tries per second, or how many seconds it spends trying an IP address.

There is a 0.1-second sleep time between attempts in the virus code, but it also sets ten seconds timeout for connect() and, very often, non-existing, non-reachable or non-listening computers will take all of this delay. When the connection does succeed, the virus switches to blocking mode and can spend, generally speaking, an indefinite amount of time trying to send() itself (well, perhaps not indefinite, but certainly a long time) and, if successful, to recv() the reply (and that really can be indefinite with blocking I/O). So, to estimate the rate at which an infected computer probes IP addresses we use the gathered statistic data. As a result of the IP address selection algorithm used by the virus, most of the HTTP requests to a computer are from the same class B network. Note, *from*, not *to* – the virus sends most often to the IP address on same class A network, but since there are 256 times more possible class A IP addresses than class B IP addresses, you are much more likely to *receive* the virus from your class B network. So, the class B attackers provide much more fodder for statistics.

Over the first 33 hours, my computer was sent the virus 374 times from my class B network (65.4.*.* – *AT&T* @Home in the states of Washington, Oregon and a small part of Northern California), but 'only' from 68 unique IP addresses (I am talking about 65.4.*.* addresses only for now). In comparison, there were only 55 attacks in total from class A outside my class B over the same period of time. Fortunately for my calculations, an IP address on the *AT&T* @Home cable network always belongs to the same computer. So, with 374 attempts from 68 different infected computers, that's an average of about 5.5 attempts from each computer over 33 hours.

This means that, on average, a computer exhausts its class B IP range (2^16 = 65,536 minus some change representing the octets it avoids) in about six hours. So, approximately 65,536 IPs in 21,600 seconds. But, remember the algorithm: the virus tries the same class B addresses with probability of only 3/8. So, the total number of different IP addresses tried is ~65,536 x 8/3 = ~174763 in 21,600 seconds.

This works out at approximately 8.1 IP addresses per second, or approximately 0.124 seconds per IP. This is with 300 threads running concurrently (since the number of 'Chinese' computers in the Internet is still many times fewer than the number of 'non-Chinese' computers, we can safely disregard the 300 vs 600 thread difference in the estimate, especially since we are looking at computers in the US North-West, where a 'Chinese' system is not very likely). So, one thread spends about 37 seconds per IP address.

We want to estimate the number of infected computers in the whole of the Internet. To simplify the calculation of probabilities, etc. we use a very close approximation and consider only the computers *not* on our class A network – that is 255/256 of the total.

The probability of a single virus thread on such a computer trying my IP at any given moment is 1/8 x 1/(~2^32) or ~2.9E-11. Over the 33-hour period it tried a total of about 33 x 3600 / ~37 = ~3210.81 IP addresses. Given that there are 300 such threads, this becomes ~963,243 attempted IP addresses per infected computer. The probability of an infected computer not on my class A network hitting my IP address in those 33 hours is approximately: 963,243 x 2.9E-11 = ~0.000028. Thus, for me to have N hits from such computers over 33 hours requires, statistically, N / 0.000028 or ~35,671 x N such computers. Within the 33-hour period there were three hits from different IP addresses not on my class A network, with the fourth arriving shortly after. Thus my estimate is between (3 x 35,671) and (4 x 35,671) infected computers. Rounding down and up to account for computational and statistical errors (and simply to arrive at beautiful round numbers) we get 100,000 to 150,000 computers infected with CodeRed.c worldwide.

About 17 hours after my first calculation, by which time many more computers had come online after the weekend, I decided to recalculate my estimate, based on newly gathered statistics. By that time, network activity due to the virus had noticeably increased and my FakeHTTP program was registering a hit by CodeRed.c every two-and-a-half to three minutes. Using the procedure described above, I estimated the number of infected computers worldwide at this time to be between 250,000 and 300,000.

It's worth noting that most home users whose computers were infected apparently did not realize they were effectively running a Web server on their *Windows 2000* computers – the computers were purchased with *Windows 2000* and the vulnerable Microsoft IIS server pre-installed. This fact became obvious when I tried browsing some of the infected computers from which the virus had been sent. The overwhelming majority had the default 'empty' Web page that is generated by the IIS if no real Web page is present.

Today's Figures

A few things have changed since my initial calculations. Many people were made aware of the virus and the security hole in their systems, so many have downloaded and applied *Microsoft*-provided IIS patches, and today I am receiving Code Red 'only' every six to seven minutes.

Also, a modification of CodeRed.c appeared – CodeRed.d. The main difference from CodeRed.c is that this variant probes genuinely random IP addresses twice as often – a random IP address is generated with a probability of 1/4 instead of 1/8, at the price of reducing the same class A IP address probability from 1/2 to 3/8.

I have calculated estimates for both CodeRed.c and CodeRed.d today. These are: approximately 70,000 computers infected with CodeRed.c worldwide, and about 100,000 computers infected with CodeRed.d. However, this does *not* mean that there are 170,000 computers worldwide infected with either CodeRed.c or CodeRed.d – the two can coexist on the same computer and in fact I did observe both of them coming to me from the same computers.

FEATURE 1

Virus Hunting in Saudi Arabia

Eddy Willems,
Data Alert International, Belgium

In my work as WildList reporter, EICAR Director Information/Press and as a Senior Anti-Virus Security Consultant for Data Alert International and Network Associates, I receive a large number of virus samples. For this reason I make sure that I maintain a good secure structure on the hard disks of my computers. I consider this particularly important on my notebook because there is always a chance that it could be stolen and because I rarely use any anti-virus product on my notebook (which would interfere with my analysis of virus samples).

As part of my work I visit many clients (companies) who have been infected with different kinds of viruses in order to perform cleanup operations. In these situations I copy samples of the viruses onto a floppy disk and transfer these straight onto my notebook. Once they have been transferred to my notebook I use the PGP package to encrypt the viruses to prevent accidental access – you never know who might get their hands on my notebook!

Later, I copy the viruses to one of the PCs on my lab network so that I can replicate the samples and send them to the WildList Organization or to other virus labs. Any new viruses I find are PGP-encrypted and burned onto CD-ROM which is stored in a special safe for which there are only two key holders.

When I'm very busy, it might not be possible for me to transfer the viruses from floppy disk immediately. In this case I carry the disks with me in a protected sealed bag. On each of the disks itself is a clear fluorescent yellow label with the text 'Virus Infected diskette … Attention – Dangerous … Don't access this diskette'.

I always try to keep ahead of potential problems and this has worked well for over ten years now. Sometimes people say to me, 'Times are changing,' and tell me that everyone is aware now of the measures necessary to protect against viruses. Maybe so, but allow me to share with you an incident that happened a few weeks ago.

Arabian Night

Part of my job as Anti-Virus Security Consultant seems to be accumulating 'air-miles'. Recently, a very large company in Saudi Arabia asked me to scope their anti-virus project for them.

After a long flight to Saudi Arabia, I stepped off the plane with the distinct feeling that it could be the start of a long evening (it was already 10.30pm). However, after passing through passport control everything appeared to be going smoothly. I was visiting the country for only three days, so I did not have much luggage with me: just one large case and one small computer bag containing my notebook, PDA, software and some magazines.

It occurred to me that there was an unusual queuing system in place. I was waiting in line with about 25 people when I was asked to step to another queue in which there were only seven people. Shortly afterwards they asked me to change queue again, this time there were only two people ahead of me. So far so good – I thought I was very lucky!

Bag Search

When it came to my turn, the officials asked me to open my large carrying case. I opened it. The officer on duty proceeded to throw around my personal things. After about a minute of browsing he asked me to close my suitcase. Then I was asked to open my notebook bag.

By this time it was obvious that the officials were searching for something in particular. Evidently my notebook was not considered a suspicious object as it was not inspected. My pack of CDs were given a cursory glance, but the officer's eye fell upon a sealed red bag. Suddenly, seeing that there were some floppy disks inside the bag, he called loudly to another officer.

Both my bag of floppy disks and my passport were ripped out of my hands and carried away under the guard of an armed security officer.

I was horrified to realise that the previous day I had visited a company with outbreaks of W32/Magistr@mm, W32/Funlove.4099 and W32/SirCam@mm, and there were samples of each of these on the disks. I was completely astonished by what was happening and I tried to warn the airport officials politely: 'Attention please, I am a computer anti-virus consultant, there could be some viruses on the diskettes. Please take care when accessing the files on the diskettes.' The only response from the officer was: 'No problem Sir'.

I tried to ask how to get my passport back but received no response other than 'Keep ahead.' So I followed the advice and found myself in the arrivals hall of the airport.

After a few minutes I found the 'pickup' who had been arranged to bring me to my hotel. I told him what had happened, and he informed me that the customs officials are constantly on the lookout for drugs and pornography. So I concluded that customs must have been searching for pornography on my floppy disks. My driver told me that I would get everything back.

No Problem

I was directed by some airport staff to a room which was surrounded by armed security guards. Inside was a man who appeared to be doing nothing other than checking diskettes and CD-ROMs. I arrived just in time to warn this guy just as I had done before. Nevertheless, he didn't seem worried about viruses. I tried again, asking him whether he had an anti-virus package installed on his system. Another 'No problem Sir' was fired at me.

From the corner of my eye I couldn't see anything resembling an on-access scanner on this man's computer system. He continued trying to access the files on my disks, ignoring the fluorescent 'virus-warnings' labels on each of them. Again I warned him against touching the files, especially if he was connected to a network. And I warned him not to check any other disks subsequently as they also could become infected by his probably already contaminated system. It seemed that I really was talking to a (fire)wall and nearly nothing came from the mouth of this humble man.

I even asked him if he completely understood the consequences of his actions and my explanations. I received one final 'No problem Sir', together with my disks and my passport. A little disconcerted by the surrounding armed security guards I hurried away from the room with a bad feeling.

Recommendations

Could I have prevented this? Maybe, but if someone says in clear English: 'Attention, your system may be infected or damaged by use of these diskettes', and if they choose to ignore this, what more can you do? Even my fluorescent yellow labels with 'Danger Viruses' did nothing to prevent this person from accessing my 'dangerous' files.

In my opinion, if you are in a position where you must monitor diskettes and CD-ROMs, you should have a good anti-virus protection in place. Better tools should be used to search for the things these people are looking for instead of just 'clicking' on everything – using the current system, simply altering the file name or extension would mean that the files couldn't be opened.

I wrote a letter to the airport authorities after my trip to explain this incident, and recommending that their officials be more cautious the next time. I have not received a response, but I hope things will change.

This tale demonstrates that even anti-virus experts can be beaten, even when the most secure measures have been put in place to prevent outbreaks like this one.

I hope that my next trip abroad will be less eventful – especially for the airport I am visiting!

FEATURE 2

Viral Solutions in Large Companies

Tomas Vobruba, AEC Ltd.
Petr J. Drahovzal, Norman Data Defense Ltd.

Anti-virus protection of large computer networks must be taken as a very complex and difficult task. Due to the massive developments and continual changes and updates in the field of viruses and anti-virus products, active anti-virus protection cannot consist merely of installing the AV program and depending upon the program to detect and clean any viruses present in the network.

This year, the PC celebrated its 20th birthday and we can say that computer viruses are roughly the same age. The world some 20 years ago was very attractive for computer virus writers, due to the first wave of the Information Technology boom:

- IBM presented an idea that the computer would have a place in every household.
- Companies started to deploy ever increasing numbers of computers.

And more was to come...

The need for data sharing between computers grew with the beginning of the computer age. At that time, connection between computers was either direct or carried out through modems, to create very simple networks. More complex networks appeared as time passed by, but (in today's terms) such networks were small to medium-sized. Such network development, though, contributed to reducing the costs of both network components and computers themselves.

Later we encountered the first modern multi-task operating systems and the commercial use of the Internet blossomed. It was thus natural that someone would probably take advantage of such a situation to start writing malicious programs. People who do this have existed from the very beginning of the computer age, but the damage caused by their viruses was limited to the local environment and did not have such significant consequences on the economy as it has today.

Company Interests

A company, which tries to be successful in the field of new technologies needs to use both the Internet and electronic mail to its full extent in order to approach its business activities more actively. As it seems only natural that viruses use electronic mail as the 'best' environment for spreading around the world, we seem to come to the very principle of today's virus problem.

It is also only natural that any company would not want to lose its credibility by loss of valuable data, and would not want to experience any financial loss incurred during the fixing of virus incidents. Such a company needs to face the current situation (the world of viruses) responsibly, and needs to deploy an active and effective anti-virus solution.

Once again, the very principle of the matter is that the anti-virus solution does not end with the deployment of an anti-virus program.

As we mentioned in the beginning of this article, it is a much more complex problem than we might think, so we need to make sure that:

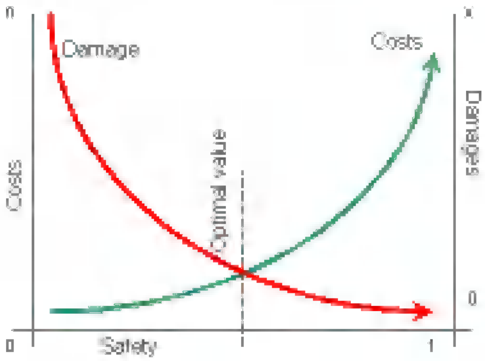
- Selection and implementation of a proper AV program must be backed by a security (software and hardware) audit.
- The needs of the company must be analysed and such needs have to be adjusted according to the current situation on the AV solution market.
- The security team must create a security policy, which will cover all possible threats related to the security of the information within the company.

We know that complex solutions, which require security analysis, project planning, complete anti-virus software and a security policy are generally very costly and time-consuming (as illustrated in the figure below). Therefore, it is necessary to find a reasonable compromise.

The following is the story of one large company in the Czech Republic which found its systems compromised by a virus infection despite the presence of an AV program on its servers.

Company XYZ

This company (let us call it XYZ) owns several thousand workstations and several hundred file servers. All of its computers are linked into the local LAN, and each such network has a connection to the central LAN. The network topology results in the asterisk structure, which is a very common structure for large networks. What was unusual in this company's network management was that the network was not centrally managed; the network had local, decentralized management. Some sort of network centralization was managed through *HP OpenView* (management of LANs, record of SW and HW, and so on).



An anti-virus program was managed locally within each of the LANs. The disadvantage of such a solution is obvious: if the local AV administrator does not follow the security policy closely (i.e. does not update the AV program on a regular basis or does not carry out regular checks of the AV program's functionality), there is a relatively high risk of virus incidents.

In fact, company XYZ did not avoid a virus incident, because of several crucial factors which allowed the virus to spread throughout the company, even though the virus had been around for more than a year.

The facts:

- Company XYZ did not have a correctly deployed (and updated) anti-virus system for both data files and scanning engines.
- The anti-virus protection was present on the company's file servers and mail servers (thus was not dependent on the PC protection only), but the virus which spread the infection came in by a channel other than email.
- The anti-virus product installed on the file servers was updated and active, but included a bug, which we will describe in the following text.

The Virus

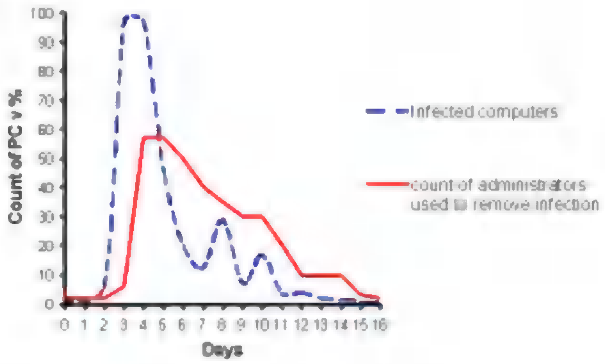
Finally, what is the virus we refer to? Probably the worst virus the network administrator could have come across. In this case, the network became infected by FunLove. FunLove spreads within the local network, looking for the files in the PE (Portable Executable) format, such as OCX, SRC, or EXE.

This type of virus, after activation of its dropper, searches through all of the local disks and network sources. If it detects any of the above-mentioned file types, and has the right to write, it infects them. Obviously, with limited or insufficient anti-virus protection, the virus will spread throughout the company exponentially. The broken line in the chart shows how fast the virus spread in company XYZ.

The virus infects all types of *Windows* operating systems. If it is run on *Windows NT* under the administrative rights, for example, it makes changes in ntoskrnl.exe and NTLDR and, after the machine restart, every user obtains local administrative rights. That way, the virus gains access to virtually any files.

As we can see, the virus can (in a very short period of time) attack and paralyse an entire computer network.

How could such a virus bypass the security policy and anti-virus protection of the email servers? The answer is (as



usual) very easy. The infection used the weakest point of every company's security: the human factor.

The Notebook Factor

An infected external notebook (without anti-virus protection) was connected to the local network and the virus distributed the infection through the files on file servers. Even though the file servers had anti-virus protection, the AV program was defective, as the software did not scan the files coming from and being changed on the local network! The infection continued to spread from computer to computer. Unfortunately, the company's technical support personnel who were responsible for anti-virus protection underestimated the threat in the first phase, and the virus was able to multiply very easily throughout the entire network to paralyse the network traffic.

The external supplier of the AV solution was not able to react instantly, because he was unaware of the problem and his assistance was not requested by XYZ until the following day. The information about the virus infection had to be tracked by the AV supplier himself, because the whole of XYZ's network was paralysed, rendering it impossible to send a sample of the virus via email. Once the virus had been identified, all of the necessary steps for disinfection could be started.

Unfortunately, it seemed that it would be impossible to stop and dissolve the fast-spreading infection without killing the network operation first! Such a solution was out of the question however, since it was vital for the company's business that at least some of the network segments stay in operation.

A Solution

The AV company needed to come up with a different solution and faced a difficult task: to find out how to get rid of the virus while allowing the network to remain in full operation, all as soon as possible.

The experts worked on the solution all day long. The cleaning process was complicated further by the fact that the virus infected workstations with an out-of-date AV program and thus the infection kept repeating. It was necessary to install a new, updated version of the AV product, which could handle the virus in the proper manner. To do this, the workstations must be virus-free. Such a condition could be achieved only if the following plan was closely observed:

- Set all sharing on the servers to read-only.
- Replace the defective AV program with the functional one.
- Clean the data content of the file servers. If the file server was infected, re-install the system and clean the data content (sharing still set to read-only).
- Create the emergency diskette with modified AV program, which would launch the cleaning process from MSDOS and start the new AV installation routine (from the installation prepared on the network drive).
- Reinstall those *NT* workstations with infected file system (NTFS). The data disks with the NTFS need to be connected to a different ('clean') computer and the data disks cleaned.
- After the disinfection of the workstations and file servers, reset the sharing to the original configuration.

It is not necessary to describe the creation of the scripts and the emergency diskette with the modified anti-virus program. We would only like to pinpoint the fact that the entire cleansing process needed to be done very quickly to avoid further costs to the company.

A total of 50 administrators were devoted to the cleansing process and they successfully cleaned the majority of computers in two days. The graph above shows the relation of disinfection to the number of devoted personnel as well as the time frame of the process.

According to the volume of scanned data, the number of computers involved in the case, and the complexity and seriousness of the problems, the entire process took almost 14 days. Some may object that this is too long, but since the entire company was up and running during the process, it proved to be a reasonable time.

Conclusions

Finally, we would like to mention that, if the local administrators had installed the properly working and up-to-date anti-virus protection, and if they had followed the security policy correctly, the virus infection would never have reached such an enormous number of workstations. The administrator could not affect the infection of data storage on the file servers; nevertheless he could have influenced the quality of the anti-virus protection of the workstations.

Therefore, it is recommended, in larger companies, to create a special team, having the maintenance of the company anti-virus protection as its primary task. 'Company XYZ', have followed this recommendation since this security incident to cut down the possibility of future virus incidents significantly.

FEATURE SERIES

Worming the Internet Part 1

Katrin Tocheva
F-Secure Corporation, Finland

March 26, 1999 marked an important milestone in the development of computer viruses, when the Melissa virus caused a global epidemic. During the two years that followed, there was a dramatic increase in the number of worms that spread via the Internet, and in particular the number of worms that use email clients. This series describes the different methods by which such viruses spread and considers the different kinds of worms, according to the environment in which they spread.

History

Science fiction writer John Brunner was the first to use the term ‘worm’ to describe a computer program, in his novel *The Shockwave Rider*, published in 1975. Brunner’s ‘worm’ was a computer program used to shut down a network of a totalitarian government which controlled its citizens. In 1979, a real worm was created as an experiment in Xerox PARC [J. Shoch]. Its aim was to perform a useful job, but as a result of bugs in the program code, it crashed all the computers connected to the local network. This experiment with a replicating code demonstrates how dangerous it is to use such programs, even with good intention.

In 1987 the first chain letter – a worm that requires human assistance to replicate, was spread in Europe and the USA. The so-called ‘Christmas Exec’ virus was written in REXX script language. In November 1988 the first widely-spread worm, Morris (aka the Internet worm), hit the network. Written to spread through VAX and SUN systems, it used security holes and paralysed the entire network in a few hours. This incident was reported in a large number of media publications and raised the important question of computer security.

Since then worms have been accepted as malicious code. Over the following years the development of boot, file and later of macro viruses took place until 1999 when virus writers found an even more powerful and faster way to spread their creations. In March 1999 the Internet was hit by the Melissa virus. As a macro virus Melissa infects *MS Word* documents, but it also contains a worm component – a routine that spreads it via the Internet using one of the most popular email clients, *MS Outlook.* Since then many new Melissa variants and other new viruses using the same method of spreading via the Internet have been discovered.

The year 2000 saw Visual Basic Script (VBS) viruses take hold when VBS/Loveletter caused a global epidemic. Later another platform, *Linux,* was targeted with creations like Ramen.worm, showing that *Linux* is not as secure as was thought [S. Rautiainen]. Recent viruses have attacked Internet Information Services (IIS) – Code Red appeared in July 2001, targeting tens of thousands of severs.

Definition and Types of Worm

Worms are computer viruses with the ability to spread through a network. Thus a worm can be defined as a program that is able to spread through a network. Depending on the type of network that is used, they can be classed in two main groups: network worms (which spread through a local network) and Internet worms (which spread through the Internet).

Earlier classifications of viruses, and particularly of worms, were made when there were only a few methods of spreading. Worms were separated mainly into three groups: Chain Letters, Host Computer Worms and Network Worms [V. Bontchev]. Today, with developments in software and applications, there are many different types of worm using vulnerabilities, security holes and the possibilities presented by powerful new languages and software. The use of the Internet has increased drastically and, with this, the possibilities for a fast transfer of data. All this has resulted in the development of several different types of Internet worm over the last few years.

Depending on the platform they use, Internet worms can be separated into three main groups: *Windows*, Unix-like (*Linux*, Unix, *Solaris*) and Macintosh. Depending on the application they use to propagate, Internet worms can be further grouped into worms that target IRC, various email clients, Gnutella, *MSN Messenger* and so on. Finally, Internet worms can be separated into groups, depending on the method they use to propagate.

Windows Worms

Windows, being the most widely-used operating system, remains the main target for virus writers. To be able to spread successfully most worms use popular *Windows* applications, others target a particular application just to prove the concept that such viruses can be written.

Internet Relay Chats (IRC) are channels (also known as chat rooms) that allow multiple users to communicate with one another in real time. mIRC and Pirch are both IRC applications. IRC worms are worms that are able to spread via IRC channels. Their method of spreading consists usually of a few steps only. First the worm tries to locate the IRC installation directory. Once found, the worm modifies or overwrites the script file (‘script.ini’ in mIRC and ‘events.ini’ in Pirch) with its own code. The spreading code usually hooks the command that will execute when the user enters a channel (join) and will send (dcc send) a previously saved copy of itself. The worm will spread further when another participant(s) in the channel clicks on the received file.

At the beginning of September 2001 the first worm that spreads a Desktop theme file was discovered. Known as Forca (also Theme), this worm is another IRC worm. Its worm code resides in a Desktop theme file and runs when the screensaver is activated, but it propagates via mIRC. To do this it uses a small routine, which is placed inside the actual theme file. It locates the mIRC installation folder and, using another file, modifies 'script.ini'. The contents of this file send a file called 'LaraCroft.theme' (previously saved in the Windows folder) to the chat room when the infected user joins a channel. As an IRC worm Forca does not use any new technique, but in spreading a Desktop theme file it represents another proof of concept.

Writing a virus code for IRC applications does not require a great deal of knowledge and such worms will continue to appear. But IRC is relatively low in popularity and not a common *Windows* application. Therefore such worms cannot be considered a great threat. Nowadays, the above-described method of IRC spreading is often just one part of the method by which an email worm propagates.

Email worms

Email worms, are worms that use email clients to spread. Most of the existing email worms propagate using email applications installed on the user's computer. The most popular such applications are *MS Outlook* and *Outlook Express*. There are rare cases in which email worms spread using *Eudora* or *Pegasus*. There are other email worms that are email client-independent. These use their own SMTP engine (W32/Sircam) or patch the original WSOCK.DLL (W32/Ska). There are also worms that use newsreaders such as *Forte Agent* (WM/PolyPoster) or instant messaging programs such as *MSN Messenger* (W32/Choke).

Speed of Spreading

A standard email worms spreads to recipients each time the user sends an email message. Worms such as W95/Ska@m and JS/Kak@m fall into this category. Email worms that spread to multiple email recipients are classified as mass-mailers. The notorious W97M/Melissa.A@mm, VBS/Loveletter.A@mm and many other email worms that have appeared over the last two years belong to this group.

Exceptions

Most email worms require the user's assistance to spread – they require the user to click on the infected file which is received as an attachment to an email message. Such worms are typical chain letters [V. Bontchev].

A smaller group of the existing email worms do not require human intervention to activate. These execute when a user reads an infected email message. Such worms include the slow email worm JS/Kak.A@m and the mass-mailers VBS/Bubbleboy.A@mm – the very first worm that was able to spread via email without opening an attachment.

JS/Kak.A@m worm spreads via the email application *Outlook Express 5.0*. To do so it uses a security vulnerability in *Internet Explorer*. Once a user receives an infected email message and opens or views it in the preview pane, the worm creates a file 'kak.hta' in the Windows Startup directory, so that the worm is activated the next time the system is started. In addition, it modifies the message signature settings of *Outlook Express 5.0* by replacing the current signature with an infected file named 'C:\Windows\kak.htm'. Thus, every message sent after this point will contain the worm. While the fix for this vulnerability has been available since October 1999, the worm is still widespread.

Sendkeys

The first attempt for a macro mass-mailer was made in the form of WM/PolyPoster.A@mm virus. This is a polymorphic macro virus written in WordBasic. For the first time, a *Word* macro virus contained a payload that sent out an infected *Word* document. As a macro virus it infects *Word* documents and templates, but its payload sends the current infected document to Usenet using *Forte Agent* newsreader (if it is installed).

The virus' payload first opens the *Forte Agent* application using WordBasic's command AppActivate. After that it uses the Sendkeys command repeatedly to carry out the following actions: it selects a newsgroup randomly from its own list (such as alt.comp.virus, alt.sex, alt.drugs etc.); builds a new *Forte Agent* message, choosing the subject randomly from 21 different subjects written in its code; writes the message body text 'WM/Agent by Lord Natas' plus some nonsensical text and later encodes this message with ROT13. It then attaches the infected *Word* document with a random name and sends it to a previously selected newsgroup.

Since it sends out one document, WM/PolyPoster.A@mm could be placed in the standard email worm category. However, since it sends this document to a newsgroup that consists usually of multiple readers, PolyPoster belongs to the category of mass-mailers.

This virus spreads to newsgroups only if *Forte Agent* is installed. This newsreader is popular, but not widespread, which is one of the reasons why WM/PolyPoster.A@mm stepped back quickly from the stage of the mass-spreading viruses.

Another reason for this is the fact that virus writers found another means by which to spread their code – a much more powerful method which was easier to implement in their creations – CreateObject, which will be described in the next installment of this series.

OPINION

Sue Who?

Raymond M Glath, Sr
Independent Security Consultant, USA
rayglath earthlink.net

In an article on MSNBC.com in August, it was reported that the State Attorney General's Office of Washington had asked Internet Serivce Provider *Qwest* to refund its local DSL customers for ten days of intermittent service caused by the 'Code Red II' outbreak.

It seems that seven Seattle-area *Qwest* customers filed complaints against the company. One claimed that his company was totally dependent upon the Internet and lost $5,000-worth of business; another claimed he 'lost the use of the Internet as a resource'.

Supposedly, when users flooded *Qwest*'s telephone lines seeking help, they were faced with a waiting time of several hours and, when they did finally get through, poorly trained technicians. (But is this a problem that's unique to *Qwest*?)

The same report states that the mediation procedure initiated by the state may escalate to a lawsuit, although that would depend on the number of people affected, the severity of the problem, and the resources available in the Attorney General's office. *Qwest* spokesman Chris Hardman said, 'Code Red was a global event. There are no plans to issue credits for customers affected by it, though we certainly apologize for interruptions in service.'

Questioning Responsibilities

This brings about an interesting question: who should be held legally responsible for these acts of cyberuptions?

- The ISP, for not securing their equipment properly?
- The developer of the operating system (*Microsoft*) for leaving a security hole in its software?
- The individual miscreant who wrote and released the destructive code?
- The security company who uncovered the hole; painstakingly developed methods to exploit the hole, and then told the world, in great detail, how to do it?
- The media, since their penchant for reporting sensationalism encourages all sorts of acts aimed at gaining publicity?
- The end user (and ultimate victim) for being naïve enough to trust that their Internet connection would be constantly operable and therefore a viable vehicle for their business?
- Those who continually and loudly espouse the 'Full Disclosure' theory as a means to strengthen computer security? (Completely describing every flaw that's discovered, and providing working exploits so that people will learn to secure their systems properly.)

Each of the above has some degree of responsibility for the outcome. However, the worm's author, even if discovered, is unlikely to have the financial resources to be a worthwhile target of a lawsuit, and the media, end users, and the full disclosure folks are each too vague an entity for a lawyer to even consider. Which leaves as possibilities for lawsuits: the ISP, the operating system developer and the security company who uncovered the vulnerability.

Sue the ISP

If bringing a lawsuit against the ISP (or other corporate victim) becomes a reality, then all you Sys Admins out there had better find a way to work 36/7 to keep your systems patched. Of course, we know that updates and patches are always safe to install the very moment they're released – *they* never contain any bugs or create any problems … right?

Sue the Operating System Developer

Microsoft, the operating system developer, is a possibility for a lawsuit. Yeah, right. We've seen how they've crumbled under legal pressure from the US government.

There are some key issues here, however. The security industry has complained for years that *MS* needs to pay more attention to security. Obviously, they cannot foresee every possible hole that might be present, but when they are shown a problem associated with a specific coding technique, they should devote serious effort to sealing all similar holes in their software without responding in panic mode to each and every one that is turned up.

In this case, the open hole was yet another buffer overrun. According to a report by Benjamin Polen at Forbes.com, 'The vulnerability is so severe that anyone with modest programming skills and an Internet connection can gain complete control over a Web server running IIS.' When you look at the description of the flaw by the folks at *eEye Security*, it is not quite so cut and dry. They had to identify and solve a number of fairly complex problems in order to make their 'exploit' work. This even included the research to determine appropriate sections of *MS* code that remained static across multiple releases. That effort was, by no means, a result of 'modest programming skills'.

Sue the Company Who Uncovered the MS Hole

eEye Security was the company who uncovered the *MS* hole and proceeded not only to publicize it in great detail, but also to release source and binary code of a working 'exploit' that could be used as a part of a worm or, at minimum, as a tutorial for exploiting the weakness. Coincidentally, this company was the first to discover the actual worm, which, after disassembly and analysis, they posted publicly too. That's right – commented source code for the worm was posted on their Web site.

eEye gleaned an enormous amount of publicity from this whole event, due to, in my opinion, a couple of brilliant publicity moves:

- One of the company's founding members is a now 20-year-old allegedly 'reformed hacker', who (reportedly) used to break into military systems under the name 'Chameleon' and was raided by the FBI in 1998. This fellow goes by the super-sexy job title of 'Chief Hacking Officer'. The press simply went crazy over that title, and repeated it at every available opportunity ad nauseum.
- *eEye* played on a well known product name when they named the worm. Their press reports state that their analysts used *Mountain Dew*'s 'Code Red' drink to keep them awake during the analysis effort, hence the choice of name for the worm. This, of course, was an interesting twist for the media as well. *Mountain Dew* reported that they were sending a case or so of their product to the company, so both parties gained some good publicity. *eEye* vaulted from being a virtually unknown company to the spotlight of numerous newspaper, magazine and TV reports, and their Chief Hacking Officer was even called upon to testify before a Congressional Subcommittee on Government Efficiency, Financial Management and Intergovernmental Relations.

More on Full Disclosure

To illustrate just how silly this can become, on 1 August Chris Taylor wrote a column for Time.com entitled 'Why Worms Like Code Red Are Good For You – What doesn't kill the Net only makes it stronger. Viva hackers!'

In his article, Taylor presented a number of interesting concepts:

- There was no malicious intent from the Code Red worm. (Presumably this makes it OK.)
- The incident created a great deal of publicity for *Microsoft,* who appeared with US Government officials in a 'heroic' role.
- The majority of business-type servers run other companies' software, and were therefore never affected in the first place.
- In the long run, this makes the system stronger through the act of resistance.
- There is such a thing as white-hat hacking.

Traditionally the AV industry has taken the stance that all virus samples, even those that are 'In the Wild', must be controlled and kept within the small community of virus researchers. Indeed, there are strict, albeit sometimes informal, 'rules' that are followed regarding the methods of distribution as well as the persons eligible to be recipients of virus samples within this community.

There have been untold numbers of discussions regarding whether, and under what circumstances, an AV researcher should create a 'new' virus in the process of carrying out research. Indeed, this has even extended to lively debates over whether the act of running an 'old' virus on a 'new' platform constitutes the creation of a 'new' virus in that platform (i.e., is the virus that is produced as a result of running a *Word 95* macro virus in *Word 97* a 'new' virus that has been 'created' by the researcher who performed the test?).

Now there is a new breed of security folks who believe that samples should be set free. Within the past few weeks it has been reported that the following 'exploits' are available for download now:

- 'AirSnort', which lets hackers grab passwords and other sensitive data being transmitted in wireless 802.11b (Wi-Fi) format.
- A hack into *Verizon Wireless*' Web site that allows unauthorized access to personal information, including customer phone numbers and cellphone usage records.

Interesting theories. Should we all teach our children to respect fire by placing their arms into a blazing campfire on their second birthday?

Back to Legal

So, the legal wrangling begins with the victim. Surely their failure to patch their servers in a timely fashion makes them liable. We should begin holding 'Duck and Cover' drills for Sys Admins if this becomes a trend.

The traditional AV folks don't figure as much in this particular foray, however it wouldn't be much of a stretch for some corporate bosses to come to the realisation that they've paid an enormous amount of money for products that claim 'complete protection', yet their systems are inoperable due to a 'virus'. I'm sure it's only a matter of time before a major lawsuit appears. The AV companies have been getting away with the old 'it's a new virus, you need to get the latest update' story for over a decade now.

Interesting times

Think of the possibilities of a high-profile case due to some virus, worm or other cyberuption, where the entire knowledge base of the Attorneys, Judge, and Jurors consists of what they've learned from the popular media. Terrifying, isn't it?

Welcome to the 21st Century. Interesting times, indeed.

PRODUCT REVIEW

Sophos Anti-Virus 3.49 and SAVAdmin 2.10

Matt Ham

As the readers of *Virus Bulletin* are famed for being eagle-eyed, I should not have to point out that the title of this review has a glaring omission: a platform. In the past, standalone reviews have concentrated on a single product on a specified platform. In this case, however, the review takes a broader view of a networked environment with all the associated Web site and deployment options available. It is intended that this should become the standard method of reviewing in a standalone context (though within the limitations of the *Virus Bulletin*'s test networks). Further details will be given later of the choices made when hardware or manpower were limiting factors.

That said, the review platform used primarily on this occasion was a *Windows NT* server, with various *Windows 2000*, *98* and *ME* clients attached to it. Exact configurations were varied over the course of the tests to maximise the number of test configurations, while keeping the physical number of machines required within reasonable limits for testing. Therefore the size of network was never beyond what would be considered very small in a business sense, and stress testing was limited as a result.

Sophos Changes

Sophos have expanded their product range considerably since it was last inspected, way back in October 1998 (see *VB* October 1998, p.17) and in addition have made some serious changes to their corporate philosophy. Gone are the days of on-access scanning performed on servers for client located files to be replaced by the more common in situ method. Gone too is the insistence that executable disinfection is a creation of the devil himself; disinfection is now offered for a subset of more common viruses. Similarly notable is the vanishing of the theory that gateway scanning is a wasteful and foolish occupation – such scanners are now an intrinsic part of *Sophos* virus protection dogma. With these changes in mind, what of the product and its assorted associated services and resources? Read on for the *Virus Bulletin* view.

The Package

The *Sophos Anti-Virus* package is smaller than it was when last reviewed and slightly less sturdy than the armour-plated box inspected at that time. In terms of appearance, little has changed in the intervening years. The contents of the box comprise, in no particular order, a mouse mat, two CD-ROMs, a small book and three slim booklets. These are, presumably, in the process of changing between two different designs, since some are in the same green-and-red colour scheme as the box, while the design of other items is more dark and brooding.

The mouse mat shares a place in the 'bonus extras' category with the small book and the first of the CD-ROMs. The 72-page, book is entitled *Computer Viruses Demystified* and is an overview of virus activity, classification and prevention, together with a list of *Sophos*' choice of the top ten viruses of all time and other useful facts. It was a pleasant relief that the information contained within the book came across as informative rather than hard sell. There are some parts of this book where hard-core pedants might be tempted to take issue with the facts presented (e.g. most of the W95/CIH.10xx descriptions are accurate, but another gives April 26 as the trigger date for all variants). By and large, however, the book would be useful for average users and a good introduction to the subject for less virus-aware administrators.

The third 'bonus' item in the package is the *Sophos* corporate video CD, *Sophos and the virus threat*. This auto-launches to provide a front-end giving access to seven themed AVIs with accompanying audio commentaries. The commentaries are available in English, US English, German, Spanish, French and Japanese. The AVI files are partially internationalised too – in cases where language-specific data is displayed rather than simply spoken there are alternative AVIs.

The contents of the video clips are of a professional quality and contain information about user education and general virus-related matters which could be useful. However, there are some quirks to be seen: in the International English version, the offices in England are described as the headquarters of the company, while in the US English version there is a swing to having a US headquarters. This is not untrue, but the US headquarters is simply the headquarters for the US operation rather than a global headquarters as is possibly the assumption intended. Such shyness about a company's origins is not restricted to *Sophos*, though it seems a little risky to put two conflicting statements like this in the same place.

Onto the more relevant contents of the box: the product CD and three associated manuals. The manuals are all installation guides, covering *Windows 2000* server, *Windows NT* server and *SAVI* and will be referred to in the Documentation section. This leaves the CD as the final and most important part of the package to be inspected.

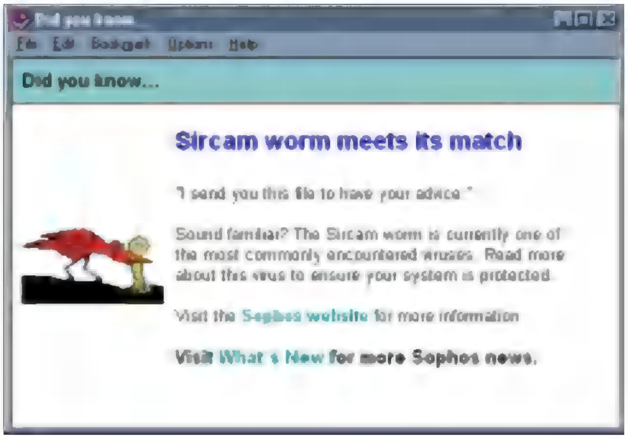
The CD autoruns, and pops up with a menu of options, which has not changed much from past reviews, together with a new pop-up 'Did You Know…' window, featuring a graphic and a small information bite from the month of release. From the main menu, though subdivided into thematic sections, the basic actions available are for software to be installed and documentation viewed.

The CD thus contains documentation and versions of the software for various platforms – DOS, *NetWare*, *Lotus Notes*, VMS, OS/2, various Unix versions. and a selection of *Windows* operating systems are all visible on a *Windows* inspection – Apple Macintosh software also exists and can be viewed on that computer. In addition to the file-based installation material there are disk sets available here for a subset of the full selection. Included are also a variety of tools, ranging from the GUI-based *SAVAdmin* to various less enormous utilities for detecting *SAV*'s status on a network. Disinfection tools are also available.

Documentation

The three guides enclosed in the box are the only printed documentation provided with *Sophos Anti-Virus* and are by no means epic in size. They are, however, easy to follow and there is no information overload. The reader-friendliness continues with the layout, which favours sparseness of text with a plethora of illustrations – if, as the saying goes, a picture amounts to a thousand words, the manuals are verbose indeed. With such a sparsity of text, there might have been a danger of vital installation information being missed out. Thankfully this does not seem to be the case – such details as how to create login scripts are covered in the *Windows 9x* client installation instructions, rather than taking such knowledge for granted.

Three guides are far from the sum total of documentation included in the package, with the CD holding a wide array of information in PDF format, and the *Adobe Acrobat Reader* also provided on the CD. These documents are arranged by language and provide a great deal more information than the printed installation guides. For example, the *Windows NT* installation guide reaches a total of 49 pages, while its manual on the CD is a more weighty 165 pages. For more detailed study of the documentation the *SAVAdmin* manual was singled out (mainly because reference to this was required more often during the testing procedure than to any other).



Did you know...

Sircam worm meets its match

"I send you this file to have your advice"

Sound familiar? The Sircam worm is currently one of the most commonly encountered viruses. Read more about this virus to ensure your system is protected.

Visit the Sophos website for more information

Visit What's New for more Sophos news.

The *SAVAdmin* manual is a document of about 110 pages – with, for some reason, the page numbering ceasing when the Glossary and Index begin. The general layout is a chapter-based format, with the contents giving subheadings beneath each chapter name, describing the main points covered within the chapter. For example, the *Configuring SAVAdmin* chapter has subheadings concerning the preference pages, configuration parameters and associated accounts. Having been a teacher in a past existence, the format of the chapters themselves struck a chord: each begins with what amounts to a learning goal for the chapter. Taking the *Configuring SAVAdmin* chapter as an example, the introduction states that it 'describes configuration options for *SAVAdmin*'. This might seem somewhat simplistic and is the only possible fault in this method of introducing information – in most cases the explanation is merely the chapter heading with a little padding.

As far as the contents are concerned, *SAVAdmin* uses a large number of self-created acronyms and standard terms which, were they explained on every occasion of their use, would increase the amount of space occupied greatly. Since the average user is most likely to dip in and out of a manual like this, an explanation of these unique parts of the language must be both comprehensive and readily available; the glossary is very much up to this task. Similarly, the combination of index and contents page make for easy location of the information required. These may seem like elementary requirements, but to my great dissatisfaction these are frequently bottom of a developer's list of priorities.

In terms of language, the documentation comes in a number of flavours. English was reviewed, though both documentation and software are also available in French, Spanish, German and Japanese in addition.

Web Presence and Support

The full range of documentation is among the selection of information to be found on the *Sophos* Web site at http://www.sophos.com/. Of the companies in the anti-virus field there is one great notable division: companies which specialise in anti-virus and those which have far wider interests. *Sophos* falls firmly into the former category, which means that virus-related information is reached as soon as the Web site is accessed – a minor advantage, but not totally dismissible nonetheless. From the homepage, where a list of recent virus alerts and a selection of press-releases make up the bulk of the on-screen information, a large number of areas can be accessed.

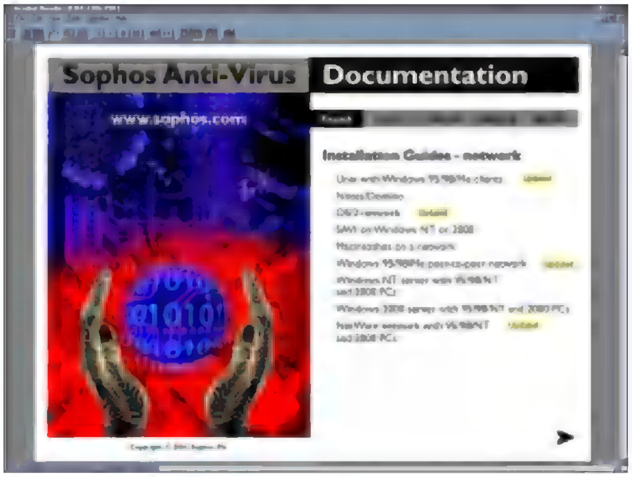
The homepage itself is optimised so as to be clearly readable on lower resolutions, leaving a rather large unused area on higher resolutions which looks a little odd. On a functional note, there are links here to the French, German and Japanese mirrors of the site, the *Sophos* mailing list for new virus-identity releases, a quick search box and a link for would-be employees of the company. A constant pictorial link in the top right-hand part of the page has changing contents upon each viewing: employment, anti-virus solutions, administration through *SAVAdmin* and the like have all been seen to feature here.

The bulk of the links from this page are to be found on the left, each of which opens a panel of further links. The link categories available are: Product Info, Downloads, Support, Virus Info, Company Info, Press Info and Partners. Of these the last three are interesting to the curious but not really relevant to the scope of this review. Product info is of note because it mentions not only *Sophos Anti-Virus* but the less well-known arm of the *Sophos* empire, that of training – which, while mostly dealing with *Sophos Anti-Virus*-related affairs, does also stray into the realm of general security matters. Given the recent number of worms whose propagation is partly or wholly dependent upon security flaws and errors, the general security arena is rapidly converging with the anti-virus industry and thus such training is not a surprising inclusion at all.

The Downloads link from the homepage offers product updates, virus identity updates and beta products directly. The latter are supplemented by research site products which are at a very early stage of development, are not guaranteed to function and thus have their own site to keep them well away from the possibility of being mistaken for a fully tested version. The use of a wide range of volunteers to test and input on the next generation of *Sophos* products can only be a good thing as far as the future of *SAV* is concerned, though having used pre-beta software from several developers in the past, I applaud those end-users brave enough to take part in such schemes.

The downloads area makes no distinction in terms of the items available between licensed and trial users of *SAV*, though the latter are required to give personal information in order to gain a time-limited download account. Updates and upgrades are discussed fully later on, but both are available here.

The full product is re-released every month and this is available as a zip, self-extracting .EXE file or a series of floppy images packaged in a *Sophos* proprietary manner.



Installation Guides - network

Unix with Windows 95/98/Me clients Updated

Notes/Domino

OS/2 network Updated

SAV on Windows NT or 2000

Macintoshes on a network

Windows 95/98/Me peer-to-peer network Updated

Windows NT server with 95/98/NT and 2000 PCs

Windows 2000 server with 95/98/NT and 2000 PCs

NetWare network with 95/98/NT and 2000 PCs Updated

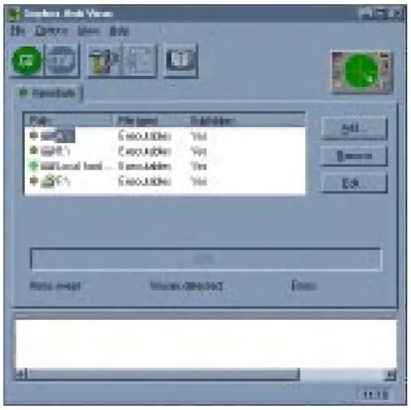
Products are in the 7 to 8 MB range for the *Win9x/ME* version, and there are no split versions other than the floppy images (something which users of slower or less reliable connections may wish to be aware of). Indeed, using a 56k modem connection proved a frustrating method of downloading, with poor data quality over a usually reliable line, though one hopes that few corporate customers will have to stoop to such depths of connection speed. Downloads over a faster connection were performed with no problems whatsoever.

The virus identity files, known as IDEs are downloadable too, and these are much smaller. IDEs can be downloaded either individually or in bulk as a zip packaged archive of those since certain release dates. The archive URL remains constant, allowing for this download to be scripted if required, and typical IDE file sizes are in the sub-1000 bytes range. Some more novel or complex viruses do require larger IDEs, though the largest currently archived of these is for PDF.Peachy at the relatively massive 16 KB, which is still a far cry from a noticeable download time on even the slowest connection.

Downloads lead on to Support on the homepage, which is quite a cover-all category in comparison with its innocuous name. 'News' is a listing of recent issues – currently almost entirely a comprehensive listing of various *Microsoft* vulnerabilities, with a smattering of more *SAV*-specific issues on a number of platforms. There are, in turn, links to the various patches for these vulnerabilities which makes this a useful resource.

The next general support category is FAQs, broadly divided into genuine FAQs, a set of disinfection instructions for several 'popular' viruses and some 'Other Articles' which are, in reality, related to the most frequent FAQs about *SAV*. Although not used particularly in the review process, the information concerning *SAV* seems very much what might be required as a mini trouble-shooting guide.

The disinfection information covers registry repairing and specific *SAV* settings for a variety of common viruses. This should, no doubt, save a great deal of time for the *Sophos* support team. The need for specific settings and registry tweaks is a common problem with many anti-virus solutions, and is becoming more of an issue with worms which affect an ever-growing number of registry entries (for a particularly impressive specimen check out the VBS/Merlin.C description on *Trend*'s site for the manual fix instructions). Finally in the support section comes the option to submit a support query. This allows a user to pick the brains of the *Sophos* Support team concerning problems with *SAV* or viruses in general. Although a judgement of support quality is fraught with problems, the general feeling on newsgroups is that *Sophos* support is of above average quality, and that its much vaunted all-day, every day support (included in the licence) is true to its name. This is in contrast to one nameless company, whose special added-value added-cost support cover for the millennium period specifically stated that holidays were not covered by



support under the provisions of the agreement made.

The remaining section linked from the homepage is the Virus info section, which is the only direct link of those discussed. Various virus and hoax write-ups are located here, together with statistics, general virus-related papers and links to the *Sophos* email notification and information feed services. The former supplies an email notification in the case of a new IDE release by *Sophos* which is considered particularly noteworthy. A common statement in these notifications is that the virus is only of note due to competitors' press releases. This leads to slightly more alerts from the service than might be expected from a noteworthy virus. On the other hand, it does have the potential to lessen panic if what amounts to a hype warning is received, thus this is perhaps a price that many will be willing to pay.

The fact that it has not been noted before draws me to the conclusion that the information feed is a relatively recent innovation, allowing *Sophos*' top ten or five virus alerts or encountered viruses for the month to be included in a third-party Web site. This might be of particular use to intranet virus resource sites, since each named virus on the lists is linked to its corresponding write-up on the *Sophos* site. This information is also available from the Press office section of the Web site in a prettier pie-chart format.

Installation

The software can be installed from downloaded files, either complete or as a floppy disk set, from a server or directly from the CD. This installation information refers to a file-based installation on *Windows 95*, though reference is made to those areas in which other platforms are different in their behaviour.

In a major departure from many other products the *SAV* installation interface is much more of a *Windows*-style GUI, matching that of *SAV* itself, rather than the time-worn InstallShield interface. If left in one state for more than five minutes the installation program provides an alert to this effect and asks whether installation should be aborted, which should ensure against accidental partial installations.

After a brief introduction screen the choice is presented of whether to install as a local or central installation, whether to install the on-access component *InterCheck* and whether to install the monitor for the *InterCheck* software.

The central installation is discussed later, and a local install considered here. After selection of source and destination directories, a final choice is given as to whether to run *SWEEP* (the on-demand component) on startup, a summary of setup information is provided and the software is installed when finish is selected. At this point the remainder of the installation takes less than a minute on any of the systems used.

When installing from a server, matters become somewhat more involved, though not particularly so for this type of installation. *SAV* must first be installed using the 'Create Central Installation' option, which results in a central repository of information for all *Windows* platforms supported in the directory chosen for this purpose. This has global attributes associated with it for installations made from this installation point, termed the Central Installation Directory (CID). These are, again, the option to run *SWEEP* at startup, enable auto-upgrading from the central copy and prevent removal of *SAV* from machines on which it has been installed.

The options here are clearly designed for situations where larger networks are to be deployed to, with possibly unwilling recipients. This theme continues as the next selection is whether the Auto-Upgrade is to be interactive, with end-user configuration control, or non-interactive with a defined set of options. It is possible to allow users to postpone auto-upgrading, or to deny this entirely.

When the CID is set up as described, installation on workstations is performed by running Setup once more from that directory, while on the machine to be upgraded. This is clearly designed to be integrated with login scripts, since individually heading to each machine would be a rather tedious occupation for even the smallest of networks.

Installation on other platforms varies in complexity. The *NetWare* NLM is simply copied to the server and executed, which will produce a fully operational installation. Updates here are as simple as adding the new NLM to an upgrade folder, which may be configured to trigger upgrades within a minute of their arrival, or to poll for upgrades on a somewhat more relaxed timescale. The Mac product is as automated as the *Windows* versions, though notable for its completely different style of interface – this being adapted for the conventions of the Apple operating system. DOS is also supported, with a DOS-style GUI making the process relatively user-friendly for that platform.

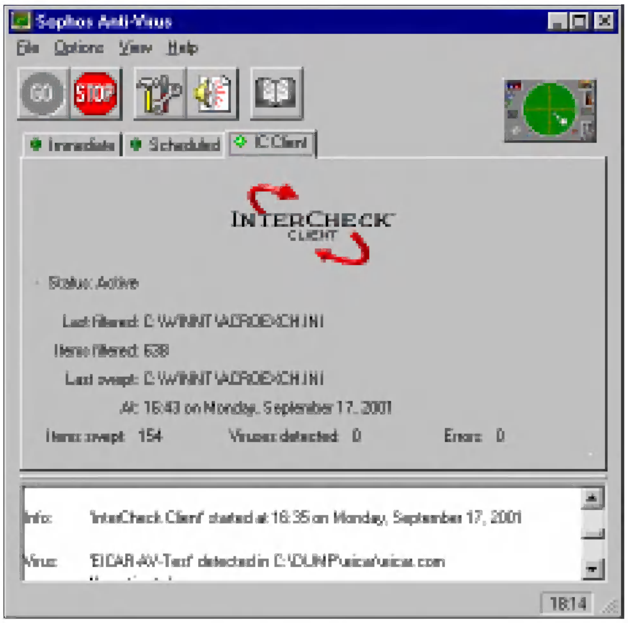
Updating and Upgrading

SAV is somewhat unusual in its upgrade frequency, there being a once-a-month complete revamp of the entire product on an internal level – though the GUI and CLI interface remains almost totally constant. Ignoring, for now, the frantic activity this must produce in the QA department, this update is distributed in two forms: as a Web resource and a monthly CD, both of which are included in the standard licence fee. To this, only virus data in the form of IDEs is added, the updates having no information but that required to add detection of the virus. One oddment of note concerning the update of virus definition IDE files is that there are several platforms on which the software must be restarted so as to institute scanning with these identities included. Notable among these are the *NetWare* and *Windows NT/2000* versions of *SAV.*

This reliance upon monthly software upgrades has positive effects on stability, since all users can be assumed to be using one of only 12 distinct versions each year, rather than the piecemeal upgrades, patches and updates available under a more traditional scheme. The downside, however, is that such matters as changing extension lists are not available through updates but only in the monthly upgrade – a matter which has resulted in several recent problems with *SAV*'s extension listings in recent comparative reviews. The sense of urgency in performing these extension changes manually is being upgraded currently on the *Sophos* Web site, but looks unlikely to become automatic in the near future.

As far as automating this upgrade and update procedure is concerned, *SAVAdmin* is currently the chief tool available for this process, though a major new product along these lines is rumoured to be under development as we speak. *SAVAdmin* is too large a product to be treated in this section and is fully investigated later in the review.

SAVI is somewhat outside the category of a separate platform, the acronym standing for *Sophos Anti-Virus Interface*. This is *Sophos*' offering to those companies who wish to make use of the *Sophos* engine within their own programs. The *SAVI* installation as described in the third manual thus does not actually do very much without a bolt-on application to make use of the functionality which it provides. Such applications are outside the scope of this review, but in general the upgrade-update process for *SAVAdmin* is very similar to that required for a *SAV* installation on *Windows NT/2000*.



Status: Active

Last filtered: C:\WINNT\ACROEXCH.INI

Items filtered: 638

Last swept: C:\WINNT\ACROEXCH.INI

At: 16:43 on Monday, September 17, 2001

Items swept: 154 Viruses detected: 0 Errors: 0

```
Info:   'InterCheck Client' started at 16:35 on Monday, September 17, 2001
Virus:  'EICAR-AV-Test' detected in C:\DUMP\eicar\eicar.com
```

Features

The basic GUI for *SAV* on *Windows* products has remained the same since time immemorial – certainly through two major revamps of the corporate identity as noted from Web site and manuals. The interface as it stands is of the standard type, with the topmost area devoted to control and lower areas to the display of selected options and results. Since few changes have been made in the last few years, the treatment of this GUI will be brief.

Scanning may be selected for the usual range of targets and usually uses the inbuilt editable extension list as a guide to which files should be scanned, though all files may be scanned if desired. Similarly, the portion of each file scanned is limited also by the default Quick Scan, in order to cut overheads, though files may again be scanned from start to finish if so desired by a Full Scan. Several of the virus samples regularly missed by *SAV* in comparative reviews are detected when these options are both selected.

From an administrator's point of view, these are configurable features, though for a user set up without configuration altering privileges, several such features are greyed out and unavailable. These are the setting of executable and exclusion lists, configuration of alerts and the location of the log folder.

There are also facilities for setting scheduled scans within the software which again are not necessarily available to standard users. These can take parameters from any other configured scan job and are thus just as flexible as a manually initiated on-demand scan. However, these do not allow for the addition of on-startup scans, these being solely controlled from the installation selected.

Changing the priority of the on-demand scanner, opting for a Full Scan, scanning inside archives and the scanning of Macintosh viruses are all under the control of the end user. Since configuration can be changed, albeit only to increase detection ability as would be hoped, non-administrator users can also opt to return to the default settings.

Currently, the *Sophos* Virus Library is available from a default installation, offering, in most cases, limited information about viruses detected by that build of the software. This is a temporary state of affairs, however, since plans are afoot to move this to a Web-based resource in the next few months. While slightly reducing the immediate accessibility of this information is not totally ideal, the saving in the amount of KB during *SAV* distributions should be appreciated and is presumably the reason behind such a move.

The *InterCheck* on-access component remains, as expected, unobtrusive and unnoticed until viral files are encountered. A general user has no effective control over the operation here, though details can be viewed as to which files have been scanned recently.

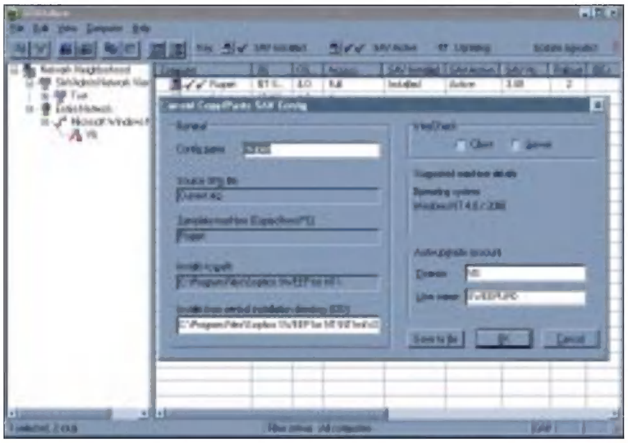
Control of *InterCheck* as an Administrator is still somewhat more limited than might be expected without resorting to configuration files. A tab allows for the setting of *InterCheck* as either on or off, but beyond this there is no apparent GUI for the alteration of these settings. These must be changed by manually edting a small configuration file. Since this file can contain detailed instructions as to the configuration of all machines involved, if it is to be rolled out on a network, this file has the capacity to become somewhat complex. Thankfully, instructions can be applied globally or over particular groups of machine, which alleviates some of this complexity, but this is far from an ideal situation. For this reason *SAVAdmin* is the logical progression in the consideration of the product.

SAVAdmin

SAVAdmin is a centralised control application for *SAV*, running on *Windows NT/2000*. It is able to monitor, implement and control *SAV* installations on both *Windows NT/2000* systems and, with some installation-time tweaking, *Windows 9x/ME*. The tweaking involved is fully supported and concerns the installation of an agent onto these operating systems, performed by a login script and is amply described in the manual. The level of control that can be exercised is considerable, and it is good to see that certificate verification is used, among other features, to ensure that the software being installed is indeed from *Sophos* rather than a cunning hacker seeking to make use of the distribution technology.

At its most basic, *SAVAdmin* allows the administrator to view the installation status of machines over the network, and to set installation policies for these machines which will be fulfilled automatically – including auto-upgrading and updating as long as the CID is kept up to date.

Somewhat convolutedly the CID in use by *SAVAdmin* may in turn be replicated onto other CIDs which might in turn be running *SAVAdmin* to control configurations on otherwise unrelated networks. This cascading system could well prove useful in larger organisations where an administrator wishes to be in charge of the day-to-day anti-virus configuration of machines in their charge, while a central repository of a guaranteed up-to-date *SAV* CID can allow them not to be concerned with downloading files every day.



Different policies may be applied to different groups – and these groups are independent of existing machine groupings by domain, though domains can be directly selected for blanket policies from *SAVAdmin*, if required.

It was mentioned previously that machine configurations for *InterCheck* have traditionally been set using a relatively cumbersome configuration file, constructed by hand. Of all the features in *SAVAdmin* which alleviate frustration, the ability to create templates from machines is perhaps the most welcome. This enables a configuration of *SAV* on a machine to be read and stored to a configuration template for later use. This template can then be used to implement installations on other machines which are identical, and thus, although machines must still be configured in the first place, changing configurations is simple once a library of templates has been created. The one unwelcome limitation of this system is that it is currently limited to machines running *Windows NT* and that *NT 4* and *3.51* are sufficiently different that files created on one cannot be used on the other. With the *Virus Bulletin* Conference around the corner, exhaustive testing was not possible for time reasons – suffice to say I've seen it in action and it operates as advertised.

Conclusions

As has been mentioned several times in this review, the core GUI of *Sophos Anti-Virus* has remained very much the same in the years since the last review but appearances and interfaces are almost the only things which have not changed. Admittedly there is a lack of the more gimmicky scanner features since *Sophos* is aimed at the corporate market, but this has the positive effect of making it stable – no crashes at all were noted in the test setups used. To the core desktop scanners *Sophos* have added a number of scanners – some not even mentioned in the text such as the forthcoming SMTP scanner. All in all, a sturdy product, with an increasing number of products widening its range and manageability, though the monolithic nature of monthly upgrades and ensuing delay in extension changes remains an irritation.

Technical Details

Product: *Sophos Anti-Virus v3.49* and *SAVAdmin v2.10.*

Developer: *Sophos Anti-Virus*, The Pentagon, Abingdon Science Park, Abingdon, Oxfordshire OX14 3YP, UK; tel +44 1235 559933; fax +44 1235 559935; Web http://www.sophos.com/.

Price: *Sophos Anti-Virus* is available on a subscription basis, with prices ranging from £10.00 per user at 1000 users to £54.00 per user at five users. Contact *Sophos* for further details.

Test Environment: As a result of the methods and nature of testing used in this review, *SAV* was run on a wide variety of machines. Space does not permit a full listing of the configurations for machines on which *SAV* was installed, though operating systems were *Windows 95*, *98* , *ME*, *NT 4.0* and *2000* with a variety of service packs and version numbers. *SAVAdmin* and central installation directory installations were performed from a Compaq Prolinea 590 with 80 MB of RAM running *Windows NT Server v4.0 SP5*.

SUBSCRIPTION RATES

Subscription price for 1 year (12 issues) including first-class/airmail delivery:

UK £195, Europe £225, International £245 (US$395)

Editorial enquiries, subscription enquiries, orders and payments:

Virus Bulletin Ltd, The Pentagon, Abingdon Science Park,
Abingdon, Oxfordshire, OX14 3YP, England

Tel 01235 555139, International Tel +44 1235 555139
Fax 01235 531889, International Fax +44 1235 531889
Email: editorial@virusbtn.com
World Wide Web: http://www.virusbtn.com/

US subscriptions only:

VB, 50 Sth Audubon Road, Wakefield, MA 01880, USA

Tel (781) 2139066, Fax (781) 2139067

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of any methods, products, instructions or ideas contained in the material herein.

END NOTES AND NEWS

COMPSEC 2001 takes place 17–19 October 2001 at the Queen Elizabeth Conference Centre, London, UK. Visit the Web site http://www.compsec2001.com/ or contact Tracy Collier: tel +44 1865 843297; email t.collier@elsevier.co.uk.

Internet Security runs from 23–25 October 2001 at ExCel, London, UK. Contact Andy Kiwankua: tel +44 20 8232 1600 ext. 246, email andy.kiwanuka@pentoneurope.com, or visit the Web site http://www.internetsecurity2001.com/.

The Black Hat Briefings and Training Europe take place in Amsterdam this autumn. Training runs from 19–20 November and Briefings from 21–22 November. For more information, as well as details of other Black Hat events, visit http://www.blackhat.com/.

The 4th Anti-Virus Asia Researchers (AVAR) Conference takes place on 4 and 5 December 2001 at the New World Renaissance Hotel, Hong Kong (see this issue, p5). For full details about the conference visit the Web site http://www.aavar.org/.

Information Security World Asia 2002 will be held 16–18 April, 2002 in Singapore. The show promises a wide ranging exhibition, discussions of the latest security issues and a number of interactive workshops. The show runs alongside Cards Asia 2002 and Mobile Commerce World Asia 2002. For sponsorship and exhibition opportunities, and further information about the show, visit the Web site http://www.isec-worldwide.com/isec_asia2002/ or contact Stella Tan: tel +65 322 2756; email stella.tan@terrapinn.com.

ITsecurity.com has reported that it receives more than 35,000 unique visitors every month. The Web site provides information on all aspects of IT security and includes news coverage, a security product database, a comprehensive security glossary and links to other security sites. The latest feature to be added is its Active FAQ which has responded to over 100 security queries in under three months. Find out more at http://www.itsecurity.com/.

***Panda Software* has teamed up with Technotrade Co., *Toshiba*'s representative in Hungary,** to ensure that all *Toshiba* laptops bought in Hungary are protected with *Panda Antivirus Titanium* which is to be installed free of charge on all new *Toshiba* laptops. For further details visit http://www.pandasoftware.com/.

Symantec Corp has announced that its ***Norton AntiVirus 2002* is the first anti-virus software to earn the 'Designed for Windows XP' logo** from *Microsoft*. For more details see http://www.symantec.com/.

Singapore and Belgium have made an agreement to warn each other about computer viruses. Under the agreement the agencies responsible in each country for distributing virus warnings will let the other know of any potential danger. Singapore's Minister for Communications and Information Technology said that viruses are 'endemic pests that are dangerous and difficult to eradicate'.

***Sybari Software* has opened new offices in Australia, Singapore and Germany**. The offices in Singapore and Australia support an aggressive sales effort planned for the Asia Pacific market under the directorship of Zubi Khawaja, formerly of Network Associates. There are plans to open facilities in Latin America in the near future. For more information see http://www.sybari.com/.

According to a report published recently by International Data Corporation (IDC), *Trend Micro Inc*. dominates the anti-virus market in server-based anti-virus software sales. According to the same report, *McAfee* dominates the market in anti-virus solutions for subscription services and corporate software solutions. Make up your own mind as to which is the more impressive achievement – visit http://www.trendmicro.com/ and http://www.mcafeeb2b.com.

FBI investigators were able to list only 55 incidences of infection by the Anna Kournikova worm in evidence submitted for the trial of the worm's suspected author Jan de Wit. The worm is estimated to have infected millions of computers worldwide when it was released in February 2001, however US investigators were unable to provide evidence of more than $166,827-worth of damage. de Wit, also known as OnTheFly, who is standing trial in the Netherlands, is to be sentenced on 27 September.

Sophos Anti-Virus is to run a two-day training course on investigating computer crime and misuse. The course will cover the mechanics of computer fraud, the forensic and legal resources available, and teach how to control evidence, ensure its admissibility, and advance the investigation and recovery process. For details including course dates see http://www.sophos.com/.